CHAPTER 9

Analysing Multi-person Timing in Music and Movement: Event Based Methods

Mark T. Elliott, Dominic Ward, Ryan Stables, Dagmar Fraser, Nori Jacoby and Alan M. Wing

1 Introduction

Accurate timing of movement in the hundreds of milliseconds range is a hallmark of human activities such as music and dance. Its study requires accurate measurement of the times of events (often called responses) based on the movement or acoustic record. This chapter provides a comprehensive overview of methods developed to capture, process, analyse, and model individual and group timing.

In a classic paper on sensorimotor timing, Stevens (1886) used a combination of paced and unpaced tapping over a range of tempos to describe what we would now recognise as characteristic attributes of movement timing. Participants tapped with a metronome set to various tempo values for a number of repetitions and then tapped on their own to reproduce the metronome tempo as accurately as possible. Stevens presented his results graphically as time series of intervals between successive responses. He showed that timing is highly adjustable but is subject to variability in produced intervals, which increases as the target interval lengthens. Moreover, he observed that the variability is not purely random but has a characteristic patterning. This includes distinct tendencies to short-term alternation between shorter and longer intervals (at faster tempos) and longer term drift around the target interval (at slower tempos).

Many papers (e.g., see Repp & Su (2013), for review) subsequent to Stevens (1886) have examined paced and unpaced finger tapping. The goals of the research include characterising influences on timing accuracy in terms of mean and variability and also understanding the nature of patterns in the variation. Although the majority of these studies has focused on individual performance, recently there has been growing interest in the relation between the timing of multiple individuals attempting to synchronise their joint performance, with the goal of achieving coherent ensemble timing (see Elliott, Chua, & Wing, 2016, for a review of this emerging area in the context of mathematical

models). Where previously the theoretical interest focused on understanding component sources of variance in the individual (i.e., timer, memory, attention, input, and output delays), the new paradigms raise questions about forms of timing linkage, including feedback correction and anticipatory adjustments, that keep participants moving together.

Stevens (1886) collected data using Morse code signal set transmission key presses (see next section for further detail). More recently movement timing study methods have ranged from switching devices such as computer keyboard keys, push button switches, resistive and capacitive contact switches to sensors such as force transducers and motion tracking cameras capable of characterising the dynamics as well as the timing of the movements. A subset of sensorimotor timing studies often involves research around timing in musical production. This research can also involve a variety of input devices, each with a unique set of methodological constraints. When using acoustic instruments, for example, additional data capture devices need to be considered, along with methods of extracting onset locations from the musical signal. Similarly, when using Musical Instrument Digital Interface (MIDI; a universal interface to a wide range of electronic musical instruments) devices, variability and latency in the system can cause issues when relaying the device's output to the participant in real-time.

Studies of timing in western music have largely focussed on the use of a piano (Repp 1995; Shafer, 1984), largely due to the simple relation between movement, note sounded, and the possibility of mechanical measurement. Similarly, these experiments are confluent with finger-tapping studies given that expert pianists tend to exhibit particularly strong timing capabilities (Keele et al., 1985; Loehr & Palmer, 2007). The piano also supports research into a range of synchronisation types such as two players following each other (Goebl & Palmer, 2009), a single player following an external stimulus (Goebl & Palmer, 2008), and a single player using both hands (Goebl et al., 2010). With both upright and grand pianos, sensors or microphones can be placed inside the instrument (Palmer & Brown 1991; Shafer, 1984) in order to record the moments at which the hammer strikes the string. More recently, electric pianos tend to be more widely used (Goebl & Palmer, 2008; Henning, 2014) due to their ability to output MIDI messages and to modify musical parameters such as playback time and timbre.

Other research has considered a broad spectrum of instrument types, each bringing challenges in terms of capturing the acoustics and defining movement timing events. De Poli et al. (1998) analysed expressivity in solo violin performances, whereas Rasch (1979), Wing et al. (2014), and Stables et al. (2014) present models for interpersonal synchronisation in small string ensembles, namely trios and quartets. Similarly, Ellis (1991) and Friberg and Sundström

(2002) investigate swing ratios in solo saxophone and percussion performances respectively.

This chapter is structured in five main sections, as follows. We start with a review of data capture methods, working, in turn, through a low cost system to research simple tapping, complex movements, use of video, inertial measurement units, and dedicated sensorimotor synchronisation software. This is followed by a section on music performance, which includes topics on the selection of music materials, sound recording, and system latency. The identification of events in the data stream can be challenging and this topic is treated in the next section, first for movement then for music. Finally, we cover methods of analysis, including alignment of the channels, computation of between channel asynchrony errors and modelling of the data set.

2 Data Capture

2.1 *Capturing Movement*

Early studies into sensorimotor synchronisation focused on a very simple motor action in the form of a finger tap (Repp, 2005). Not only is this a simple action for most participants to perform, it is also an easy event to record. When people produce a finger tap action, there is an asymmetry in the flexion and extension segments of the movement (Balasubramaniam, Wing, & Daffertshofer, 2004). This results in a short impact time of the finger onto the surface, generating strong tactile feedback (Balasubramaniam et al., 2004; Elliott, Welchman, & Wing, 2009a) that participants align with the external beat. By recording the impact time of the finger, researchers subsequently have an accurate event onset time of each finger tap. This is how one of the earliest known sensorimotor synchronisation experiment was implemented (Stevens, 1886). Participants tapped their finger on a Morse code key with the electrical contact recorded on a smoked drum kymograph. On a kymograph the timing is measured from distances between pulse marks on the surface of a drum rotating at constant velocity.

The modern equivalent of Stevens' (1886) approach is to use some form of touch sensor connected to a computer. The times between movements are determined by reference to distinct events registered by the sensor. Force sensitive resistor (FSR) materials are particularly useful for registering finger taps (e.g., Elliott, Wing, & Welchman, 2010; Schultz & Vugt, 2015). In addition to being very low cost, the sensors come in the form of a thin membrane, meaning that there is no 'travel' when the finger hits the surface (as might be the case if one used a button press or keyboard to record events).

Similar devices include piezo-electric sensors and the more recent capacitive sensing technology (as used on modern touch-screens). While low cost and practical for recording the impulse response of the tap, the aforementioned sensors tend not to be sufficiently linear for measuring the amplitude or shape of the signal. In scenarios where these parameters are of interest, a force sensor (e.g., ATI Industrial Automation; http://www.ati-ia.com/) can be used (Elliott et al., 2009a).

Interfacing these sensors to a PC for recording responses usually requires a data acquisition card (DAQ). These devices capture the analogue signal from the sensor and convert them into a digital value for import into Matlab or similar software. DAQs, such as those from National Instruments, Measurement Computing and Labjack have a wide price range, depending on number of channels, maximum sampling rate and the number of functions the device has. A key advantage is that the devices can be used to output the external cues and also trigger any other external devices, so all data is both output and recorded with a common time base, i.e., synchronised. Time resolution depends on the sampling rate, but it is possible to achieve very reliable and consistent event timings from these devices.

The close relationship of sensorimotor synchronisation research to musical contexts has meant that often MIDI equipment has been used to record participant responses. In particular, drum-pads have been used as an effective tapping sensor (Manning & Schutz, 2013; Pecenka & Keller, 2011), providing a large surface area and no movement in the surface itself. Keyboards have also been used (Goebl & Palmer, 2008; Keller, Knoblich, & Repp, 2007), however the time difference between the finger hitting the key and the key travelling down to hit the sensor adds an uncertainty as to when the event onset actually occurred. There is also a level of time lag and variability in MIDI communications between devices and the computer software. This has been identified as a small but not insignificant amount of delay (Repp & Keller, 2008; Schultz & Vugt, 2015) and, hence, should be characterised and accounted for when using this interface for timing experiments.

2.2 *Example of a Simple, Low Cost System for Recording Finger Taps to Auditory Cues*

Both the sensors and hardware for collecting data from tapping studies can range from very high-cost (e.g., force sensors with a high specification data acquisition card) to low-cost (simple impulse detecting sensor, with sound card input). Figure 9.1 provides an example of a simple solution that can be applied in fieldwork to record one or more participants performing a tapping experiment.

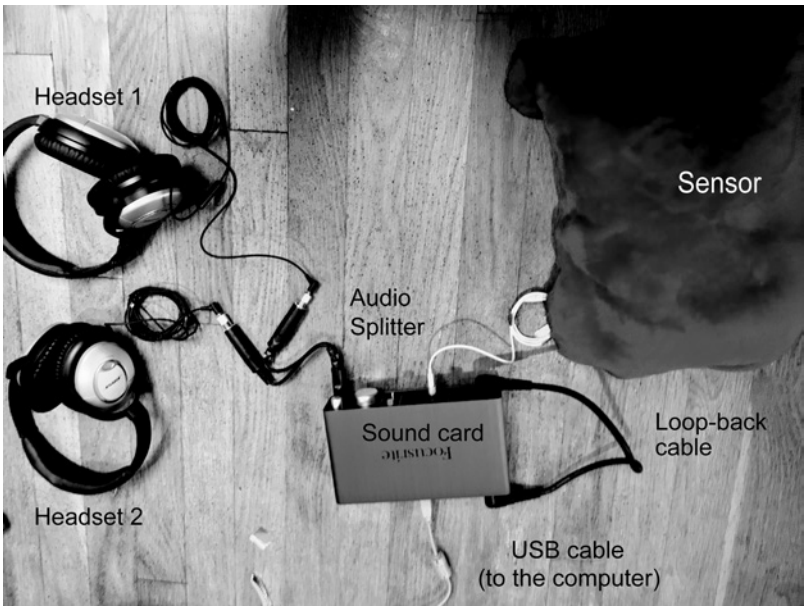

FIGURE 9.1 *Example of a simple, low-cost experimental setup for capturing finger tap responses from group timing experiments.*

One or more sensors are connected to a sound card. Sensors can be as simple as a wooden box or a soft pad in which earbuds are installed. These may be used as very cheap but low-sensitivity microphones that are well suited to record a direct touch on the surface to which they are attached, and are insensitive to external noises. This setup is able to detect even a light touch, if the soundcard is set to a high gain.

An external sound card is connected to a computer via USB. The computer sends an auditory stimulus (such as a metronome) through Digital Audio Workstation (DAW) software such as Cubase or through designated software such as MatTAP (Elliott, Welchman, & Wing, 2009b; see Section 2.6). A loopback cable is installed so that stimulus and output are recorded with zero latency, and the responses and loopback stimulus are recorded on two or more separate channels. One or more high-quality headsets are connected to an audio splitter so that participants can hear the stimulus. If participants synchronise only to one another, the headsets can be removed and sensors can be made of hard resonating material such as a wooden box.

2.3 *Complex Movements*

It is clear why finger tapping became the *de facto* task for early sensorimotor synchronisation experiments: simple equipment setups can be used

to accurately measure the event onsets for timing analysis. More recently, researchers have been interested in increasingly complex movements. For larger groups performing more complex movement interactions, data involving temporal and spatial dimensions must be recorded. This could be upper limb movements such as choreographed ballet movements (Honisch, Roach, & Wing, 2009), or lower limb movements such as walking or bouncing (Georgiou, Racic, Brownjohn, & Elliot, 2015). There are two ways of typically capturing these movements. The first is using marker-based motion capture and the second is using inertial measurement units (IMU). Here, we provide some recommendations in their use in the context of movement timing experiments.

2.4 *Video Motion Capture*

3D motion capture systems such as Vicon (Vicon Motion Systems Ltd, UK) and Qualisys (Qualisys AB, Sweden) are considered to provide the gold standard in terms of accuracy. Movements can typically be captured with an error of <0.5mm at hundreds of samples per second. On the negative side, systems tend to require calibration over a specified capture volume at the beginning of each session and reflective markers must be accurately placed on the bony landmarks of the participant's body. Furthermore, post-processing can be a tedious task in terms of labelling markers for each trial, such that trajectories can be identified for analysis. While most software packages associated with these systems have an 'auto-label' feature to identify markers, this is liable to fail. Labelling can be become particularly complex for multi-person studies. At the start of an analysis, the researcher is presented with a cloud of unlabelled markers in 3D space. Markers in successive frames must be linked to define trajectories, which can be identified with body segments of each participant. To help identify individual participants in a group it is often advantageous to add extra markers (not used for tracking) somewhere on the body that is a unique formation for each group member. For example, marking out the corners of a small square on the back of Participant A, versus the corners of a triangle on Participant B can help identify which person is which during labelling. There are 'active marker' systems, where the marker itself is electronic and hence can be pre-assigned a label or ID. An example of this type of system is the Polhemus Liberty (Polhemus, USA), which uses active markers in a magnetic field to track motion.

For event based timing analysis, one is often only interested in the temporal aspects, even for complex movements. Therefore, a small number of markers can be used, rather than a full body marker set (the Vicon Plug-In-Gait marker set is in the region of 40 markers per person). It is important to choose a marker location that will provide the primary trajectory for analysis. This might be a marker on the finger for upper limb movements, or the heel for lower limb

movements. However, the marker must always be in good view of the cameras, minimising the chances of occlusions. Additional markers can be applied to other areas of the same limb or body segment, both to aid identification and also as a secondary trajectory source in case there are problems with the primary trajectory data (a consistent trajectory location across participants should be maintained however).

Trajectories for each marker can usually be output from the software as a text file, with each marker having an individual X, Y, and Z coordinate at each time sample. We provide example code for parsing these text files, using a representative output from Vicon Nexus software (see this book's GitHub repository).

2.5 *Inertial Measurement Units*

IMUs consist of two or three sensing devices. Two-sensor devices consist of an accelerometer, that measures acceleration (in units of m/s^2 or g) and a gyroscope that measures rate of angular rotation (in units of radians/second or degrees/second). Three sensor devices have an additional magnetometer included (units of Tesla or Gauss). Recent devices output values from each sensor in 3 axes. IMUs use a local coordinate system, so it is not easily possible to infer the location of a device in global coordinates. That is, if there were two devices attached to a person, it would not be possible to directly calculate the relative distance between those devices (unless the starting positions were known). Additional data fusion algorithms allow the advantages of three sensors to be combined such that accurate motion analysis can be achieved. Without these algorithms, trying to infer the positional trajectory of movement from accelerometer data alone (by integrating the data twice) will result in drift and inaccuracy from the true position. However, for measuring timing of movements (rather than position) the associated drift is not such a big issue as the timing in the data remains intact. With some initial alignment of the data with video, it is possible to identify the peaks and troughs in the acceleration data that relate to key parts of the movement cycle (e.g., walking). Alternatively, integrating to velocity can produce a clean, and easier to interpret signal, by applying both low and high-pass filters to the data.

An IMU's main advantage is that the participant is free to move around without restriction. There is no capture volume as with video motion capture and occlusions are not an issue due to the sensors being within the device. Participants 'wear' one or more of the devices on the body and are then free to move naturally. This is particularly useful for gait analysis: in a video motion capture gait lab, only a small number of gait cycles can be recorded within the capture volume. With IMUs, the participant can complete a long walk or even be recorded over a full day, dependent only on the on-board memory of the

device. A number of IMU companies have seen the potential for this and developed software alongside their devices that includes the algorithms to do a full or partial gait analysis. Current examples include APDM (APDM Inc., US; www.apdm.com), XSens (XSens Technologies B.V., NL; www.xsens.com), GaitSmart (ETB Ltd; www.gaitsmart.com), Shimmer3 (Shimmer Sensing, www.shimmersensing.com), and BTS (BTS S.p.A, IT; www.btsbioengineering.com). All but BTS use multiple devices worn on the body. The actual gait parameters provided vary amongst the software packages and many still struggle to provide accurate step length measures due to the drift issues mentioned above.

A particularly useful feature of the APDM and Shimmer devices is that they are wireless, time-synchronised devices. They come as a set of IMUs to be fitted on different body segments of a single individual. However, the software also allows raw data access such that each device could instead be fitted to separate individuals, with their activity recorded wirelessly. Given the resulting data is time-synchronised, this is ideal for group timing studies (Georgiou et al., 2015).

2.6 *Dedicated Sensorimotor Synchronisation Software*

The main challenge with setting up the data acquisition and cue presentation for both single- and multi-person sensorimotor synchronisation experiments is in minimising timing uncertainty. Multi-tasking operating systems, such as Microsoft Windows, imply that executing commands is an asynchronous process. That is, you might run a segment of code which outputs a cue stimulus every 500 ms, but the operating system will not necessarily execute that command immediately if it is busy dealing with another application in the background. This can create jitter in the cue generation, so that a stimulus that should occur exactly every 500 ms might instead execute on average every 500 ms, with actual intervals produced varying around that value (e.g., 490, 515, 516, 502 ms etc.). If the standard deviation of these intervals becomes relatively large then the impact on the analysed movement timing results will be significant (Repp, 1999). Interval variance will increase as participants correct their movements to remain in time with the varying beat. Asynchrony variance will also be artificially inflated as both the variance in the movement and the cue sum together. On the other hand, controlled manipulation of cue jitter can be effective for investigating cue reliability effects (Elliott et al., 2010; Elliott, Wing, & Welchman, 2014).

Similar issues occur with capturing responses. If a participant is required to tap a key on a standard PC keyboard in time with the beat, it is difficult to reliably record the onset time due to lags in the operating system servicing the event. Therefore, when designing an experimental setup, minimising lag time

and jitter in the cue and response signals are a key consideration. Because of this, a number of software toolboxes have been developed to assist with data capture in sensorimotor synchronisation (SMS) experiments. These include, FTAP (Finney, 2001), one of the earlier toolboxes written for Linux. This toolbox interfaced directly with MIDI instruments and allowed accurate control of cue timing and recorded finger tap responses. Max/MSP (Cycling '74, USA) is a commercial software package that provides a visual programming interface with full MIDI support. This has often been used in timing experiments with the visual programming interface allowing relatively simple onset detection and analysis to be set up with minimal coding skills. As previously mentioned, MIDI suffers from lag and some jitter. These are substantially smaller and less significant than trying to use traditional programming approaches and PC hardware interfaces, which must still be characterised. A recent study (Schultz & Vugt, 2015) has characterised both Max/MSP and FTAP experimental setups for a sensorimotor synchronisation experiment, finding the mean lag (with respect to an FSR sensor response) to be 15.8 ms and 14.6 ms, respectively. The standard deviation (jitter) of the lags was 3.4 ms and 2.8 ms, respectively. The study contrasted the two MIDI setups to a novel hardware setup using a low cost embedded controller (Arduino, www.arduino.cc). By using the Arduino device (taking the signal processing away from the PC), the lag was reduced to 0.6 ms with a jitter of 0.3 ms.

It is clear, therefore, that moving the signal processing away from the PC to dedicated hardware such as an embedded controller or a data acquisition card (e.g., National Instruments, Measurement Computing) is a good way to get an accurate cue presentation and corresponding response times. This philosophy was used to develop another sensorimotor synchronisation toolbox. MatTAP (Elliott et al., 2009b) uses data acquisition hardware interfaced to the MATLAB programming environment to provide a comprehensive toolbox that offers virtually no lag or jitter in the signal output and response capture. By using a loop-back method (see Figure 9.2), both the output signal and response can be sampled under a common clock at very high sampling rates (e.g., 10kS/s) allowing highly accurate measures of asynchrony (see Section 5.2). The toolbox further uses a graphical user interface that allows the user to accurately control cue presentation, store data and run analyses. We have successfully interfaced the toolbox with both accelerometer devices (APDM Opal) and video motion capture (Qualisys) to allow accurate measures of group movements to an auditory metronome or visual cue. The downside to this high level of accuracy is increased expense, with both the hardware (data acquisition) and Matlab (with appropriate toolboxes) adding up to a relatively high cost compared to other solutions. Regardless, much of the code we provide with this chapter has

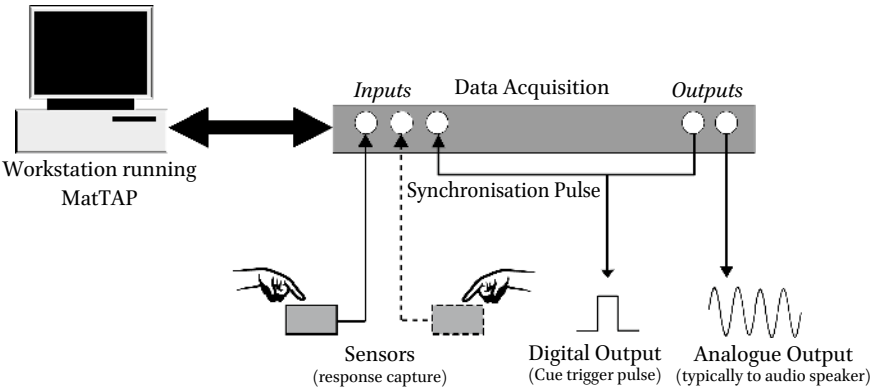

FIGURE 9.2 *A typical hardware configuration for cue generation and response capture using MatTAP (reproduced from Elliott et al., 2009b). The toolbox uses data acquisition hardware to achieve a high level of timing accuracy. (1) Two outputs are generated to drive stimuli. One is an analogue waveform, typically used to drive an audio speaker, but can also be applied to haptic or visual devices. The second output is a digital square pulse, which can be used to trigger bespoke stimuli equipment. To increase accuracy further, this pulse is fed back into the system to compensate for processing delays in the hardware. (2) Up to two sensors can be utilised to capture responses. Any sensor that produces an analogue or digital voltage, typically in the range ±5 V can be used to record events. (3) The output signals and corresponding responses are captured and stored automatically in individual, sequentially numbered .mat format files.*

evolved from MatTAP, generated from the requirements of new experiments within our labs.

3 Audio Capture

In musical timing research, both single and multi-agent studies generally adopt a similar methodology. Based on the study's objectives, selections are made for instruments, number of players and source material. Environmental constraints such as visual and auditory occlusion are then set and trials are implemented to acquire data. The performance data can be captured in a number of formats ranging from acoustic waveforms, MIDI messages, sensor-data taken from IMUs (see Section 2.5), movement data from a video motion capture device (see Section 2.4), or a combination of these systems. In this section we concentrate on the capture of the audio signals, rather than the movements of the agents producing the music. This involves recording the acoustic waveform and configuring the system to effectively derive the relevant events.

3.1 *Material Selection*

The selection of performance material is generally based on the extent to which the piece enforces performance characteristics. In synchronisation experiments, long passages of concurrent isochronous events (equally spaced notes performed at the same time) are desirable, often limiting the pool of recognised works. For this reason, Moore and Chen (2010) opted to use an excerpt from Shostakovich's String Quartet Op. 108, No. 7, which included 260 events performed in quick succession by two members of a string quartet. Furthermore, all notes are generated by individual bow-strokes, and are rhythmically partitioned into groups of four. Similarly, both Wing et al. (2014) and Stables et al. (2014) used an excerpt from Haydn's String Quartet Op. 74 no. 1, which consists of 48 x 8th notes performed continuously by almost all members of the ensemble. For studies investigating phase relationships such as Shaffer (1984), multiple voices with independent subdivisions are desirable, leading to the selection of an excerpt from Chopin's Trois Nouvelles Etudes. Specially composed pieces are also commonly used in timing studies, typically when there is a requirement for tractable context and specific musical conditions. This is the case in Goebl & Palmer (2009), where the content is easy to perform and subdivisions vary between players. This allows trained musicians to easily perform the experiment with no pre-requisite knowledge of the content.

3.2 *Sound Recording*

In some cases, it can be impractical to capture event-based performance data such as MIDI due to the acoustic properties of the instrument, or the physical restrictions that controllers impose on a participant. An acoustic violin, for example, produces notes with legato (i.e., in a smooth continuous manner, without breaks between notes) and has a small area of sound propagation. This means it is difficult to incorporate a MIDI device into the instrument without restricting the movement of the musician. This often introduces a requirement for audio recording, followed by post-processing to perform onset detection in order to derive a symbolic representation from the captured acoustic data. For music listening, instruments are typically recorded by placing microphones at acoustically relevant locations around the source and surrounding environment, with the intention of achieving a desired aesthetic. This can differ from analytical recordings where the aim is to isolate signals and derive an accurate representation of the performer's onset locations via further signal decomposition. For well recorded monophonic signals (e.g., solo instruments) comprising homogeneous fragments of sound, timing data can be extracted more easily when compared to polyphonic signals (e.g., multiple instruments played by a group) or those contaminated by noise. For this reason, close-miking

techniques in which the microphones are placed near the sound source (e.g., the strings) are generally preferred to room or ambient miking in order to obtain a higher signal-to-noise ratio; in the case of multi-person performance, at least one microphone is used per source, thereby minimising acoustic bleed. The multi-microphone setup is also advantageous in that further separation of the sources can be achieved through cross-analyses. Here, the sound pressure level of each source will vary across microphones, allowing for the attenuation of other instruments in the recording. In the case of polyphony, it may be necessary to first employ source separation algorithms such as those proposed by Vincent (2006), and then perform onset detection on the resulting streams. In the latter case, noise-reduction can be used as a pre-processing step.

For close-miking, clip-on condenser microphones such as the AKG C519 range (similar to those used in Polak & London, 2014), provide high sensitivity and greater frequency and transient responses than dynamic microphones. Whilst omnidirectional microphones can be used for this task (sound from all directions is captured with equal sensitivity), localised polar patterns such as cardioid and hypercardioid (sound in front of the microphone is recorded with higher sensitivity) are preferable as they mitigate sound capture from external sources. During this process, careful miking techniques are necessary to gain proximity to the instrument and therefore achieve a high signal-to-noise ratio. When using the close-miking technique, the microphones tend to be placed on the areas of the instrument that don't dampen the sound or prevent any free-flowing movement. In string instruments such as violins, violas and cellos, the clips are often located on the bridge of the instrument. For percussive instruments such as drums and cymbals, the microphones are clipped to the rim or to stands, so as not to interfere with the skin or plate. For brass instruments it is difficult to avoid the resonant surface of the instrument, so microphones tend to be clipped to the bell, with the microphone located inside or near to the opening.

An alternative to close-miking instruments is to use a vibrational pickup, in which a transducer reacts to vibrations of the instrument's surface material. These tend to be less common as they have poorer transient responses, but can be used when close-miking is not plausible or susceptible to noise, such as in large ensembles.

For field recordings such as those in Polak and London (2014), portable recording devices such as the Roland R-4 or the Tascam DP006 can be interfaced with microphones to capture the signal. These generally record data to an onboard hard disk or portable storage device, which can later be transferred to another machine for analysis in an uncompressed format. The main benefit of these devices is that they can record multiple channels (usually up to 6) without the need for additional computational hardware, however they have

limited auditory feedback options and recording length is often dependent on battery life. More commonly, an external soundcard with a desktop or laptop computer is used, in which the soundcard connects to a host machine via USB or Firewire. Soundcards support varying numbers of inputs and outputs and generally have assignable sampling frequencies (often set by default to 44.1kHz) for use during analogue to digital conversion. In this case, a software interface is also required to record the inputs to disk, which can be done using a DAW such as Logic Pro, Ableton Live, Audacity, Reaper or Cubase, all of which share a similar multi-track interface, with varying levels of control over the audio signal.

3.3 *System Latency*

Due to the computational overhead involved in reading, writing and processing a large number of samples each second, audio processing systems incur a time lag, known as latency, at numerous points throughout the processing chain. Furthermore, this latency is shown to exhibit high variability and information loss (Wang et al., 2010) when systems are subject to high processing loads (e.g., when multiple channels are being used to record a large ensemble), thus leading to unreliable playback. For this reason, it is generally not recommended to feed the system output back to participants via headphones when musical timing is being measured, as latency will create negative recurrent effects on the performer. In isochronous rhythmic sequences, the threshold for perception of delay is observed by Friberg and Sundberg (1995) to be around 6 ms for tones with relatively short intervals, and periodic timing correction to the delayed stimuli is observed to occur at time lags of as little as 10 ms (Thaut et al., 1998). Further to this, the standard deviation of inter-onset intervals (IOI, time between consecutive onsets) in performed rhythmic sequences is widely accepted to increase with auditory delay time (Pfordresher & Palmer, 2002). This suggests that even minimal system latency (observed by Wang et al., 2010 to be around 19 ms for Audacity with Mac OS X 10.6, when running under low computational load) is likely to impact the validity of results. If no other options are available, the signal path can often be configured to route the analogue signal directly to the headphone output, bypassing the processing chain and minimising latency caused by play-through.

If processed auditory feedback is unavoidable, such as in experiments where participants will be played manipulated versions of their input signals, the buffer size of the host software should be reduced in order to reduce the latency time in the system. This limits the time allocated to the system to process the audio samples, thus allowing the signal to reach the playback device in a shorter time period. The buffer size can often be controlled via the DAW, and can be set experimentally between 32–1024 samples. Whilst lowering the

buffer size reduces latency, it also puts strain on the computer's CPU as the processor is then required to complete more operations in a shorter period of time. Negative effects of excessively lowering the buffer size include the addition of audible noise to the signal path caused by loss of information. The input/output latency of the system also has implications for the use of external stimuli during experiments. Given the time delay incurred during playback, it is recommended to record audio signals generated by the computer back into the system whilst recording performers. This means any pre-recorded accompaniment or metronome tracks should be captured using headphones and a microphone in order to limit computational asynchronies caused by the variation in system delay at multiple points in the processing chain.

4 Onset Detection and Analysis

> A bold onset is half the battle.
>
> GIUSEPPE GARIBALDI

One of the key challenges in post-processing for event-based analysis (both for movement and audio signals) is accurate onset detection. This step needs to be applied to both the cue signal and the participant's responses. In this section, we cover the key approaches used to achieve accurate onset detection. The three main stages of onset detection are shown in Figure 9.3 and are common in both movement and music data analysis. However, completion of each stage often requires a specific approach, based on the origin of the signals.

4.1 *Extracting Movement Onsets*

In movement, onsets correspond with physical events, (e.g., the peak pressure applied to a point, a finger tap on a surface, or a sudden change in motion as measured by position, velocity or acceleration). Reliable onset detection is vital for analysing sensorimotor responses (Elliott et al., 2009b) by allowing accurate measurement of the asynchrony between the cue and the corresponding motor response.

FIGURE 9.3 *Key stages involved in onset detection.*

In the example shown (Figure 9.4), a finger tap onto a surface is captured by a force sensor, converting the force into a voltage output. This shows the *baseline,* which represents the signal prior to the finger making contact with the sensor. A rise in the signal from *baseline* identifies the *onset of attack*. The *attack* represents the rise of energy in the system from the prior state, i.e., the initial impact of the finger onto the sensor. *Peak attack* occurs when the finger reaches maximum force onto the sensor. The *onset of decay* indicates the beginning of *evanescence*, i.e., the return to *baseline* as the finger begins to lift again, off the sensor surface. For movement onset detection, it is usually the *onset of attack* or the *peak attack* that is identified as the onset time of the signal.

4.1.1 Movement Data Pre-processing

Pre-processing is the transformation of raw data to facilitate processing by the onset detection function (ODF). The first step in pre-processing is experimental design; facilitating the optimal capture of data and encoding the movement. The experimental hardware must have a sampling rate of sufficient magnitude to capture the movement without aliasing. The sensor's rise time and evanescent should be an order of magnitude faster than the movement. The magnitude of the onset sought should be readily distinguishable from that of the noise of the recording system, and distinguishable from common artefacts. The experiment should ideally offer a dedicated input channel for each element of participant response of interest, i.e., one touch sensor per finger, or a marker for each limb.

Algorithmic pre-processing addresses practical flaws in movement already captured. Low frequency human motion (below 10–50Hz) is generally

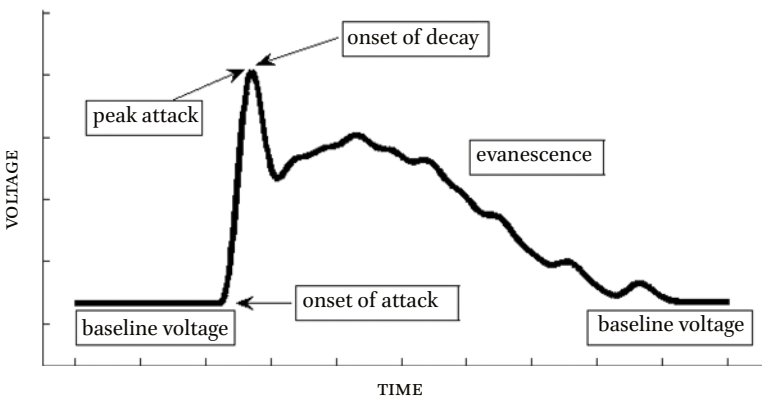

FIGURE 9.4 *Example sensor output signal resulting from a single tap.*

contaminated with higher frequency noise. This noise comes from such sources as non-ideal sensors, the environment (e.g. 50/60Hz mains hum, participant's heartbeat) and the harmonics of movement itself. The frequency components of interest are commonly emphasised via low-pass filtering which attenuates all frequency components of a given signal above a cut off value, optimally without adding delays or distortion. Classically, a low-pass filter known as a zero lag 4th order Butterworth practically implements these requirements (Winter, 2009). These filters can be implemented algorithmically in software such as Matlab, rather than requiring a hardware implementation to be used. Following the application of the filter, signal peaks of interest should remain prominent whilst noise peaks should be reduced. For onset detection, a heuristic cut off frequency can be determined by visual inspection and iteration. Note that by filtering, relevant information in the signal can be altered or lost. For temporal studies in particular, it is important to use a zero-phase filter. This is a special case of a linear-phase filter which avoids any frequency-dependent lag. The *filtfilt* command in MATLAB applies a filter both forwards and backwards, cancelling out any phase effects of the filtered signal.

To facilitate filtering, data recovered from non-ideal sensors must be sanitised. Such data should be continuous, machine-readable and exhibit values that readily allow for computation (e.g., numeric values within the maximal and minimal machine accuracy limits). *Numerical sensor artefacts* such as those arising from sensor dropouts, misconfigured apparatus etc. may return numerical error codes, missing values (e.g., empty set [] or NaN) or default values (e.g., zero). These values have no useful relation to the effect being measured and must be excluded to maintain the integrity of any analysis. Hence, data exploration and visualisation, i.e., a check upon the sanity of the data, should always be a first step.

For systems such as the Qualisys and Vicon motion tracking software, there are explicit functions that allow data for missing markers to be approximated. For less integrated systems, MATLAB functions such as *isnan, isempty, isnumeric* can be used to find invalid, non-numeric values in time-series data.

Numerical sensor aberrations include sensor saturation (where the recorded movements exceed the capacity of the sensor to report), sensor drift, warmup trends and battery exhaustion, power bounce, and other artefacts of the recording. The values may have some relation to the effect being measured, but have been transformed in a fashion not shared by the rest of the data, and hence may decrease the integrity of any analysis. These effects can be ameliorated by initialising each experimental session with a brief test run with real time sensor feedback. This will reveal aberrant values, allowing action (replacing

batteries, adjusting sensor gain). As noted previously, thoughtful experimental design is the most essential pre-processing step.

The experimental artefacts, listed above, are distinct from *participant artefacts*, in which the participant offers responses outside of those anticipated but still within the scope of measurement; ambiguous touches, mistaken taps and involuntary movements etc. These should not be removed in pre-processing, which attempts to faithfully relay participant action to the ODF. Participant outlier artefacts are treated with rigour in Section 5.3.

4.1.2 Onset Detection Functions (ODFs)

The ODF renders clearly the presence of attacks within the original signal. In musical onset detection this is often called the *Reduction step*, where the sound signal is traditionally downsampled to a 'low' sample rate (e.g., hundreds of hertz (Dixon, 2006)). However movement ODFs typically eschew downsampled.

There are many varieties of ODF: time and frequency domain, probabilistic and machine learning (Bello et al., 2005; Dixon, 2006; Eyben et al., 2010). In the context of movement, we focus on the time domain methods. In many sensorimotor studies the end of the attack, i.e. the peak of expressed force, can be considered the *intentional* onset of response. In sensorimotor timing, onsets might include peak velocity (Pelton, Wing, Fraser, & van Vliet, 2015), acceleration (Honisch, Elliott, Jacoby, & Wing, 2016), or even higher derivatives such as jerk (Balasubramaniam et al., 2004; Elliott et al., 2009a).

For attacks that are obvious to an annotator, i.e., large increase in voltage amplitude, such as transduced force in tapping experiments (see Figure 9.3 above) a simple envelope follower can be used to algorithmically extract the peaks of attacks (Eq. 1).

$$E_{(\mathrm{n})} = \frac{1}{N} \sum_{m=-N/2}^{(\mathrm{N}/2)-1} \left| x\left(n+m\right) \right| w\left(m\right) \tag{1}$$

in which $w(m)$ is an N-point smoothing kernel centred at $m = o$. This can be extended to use of the derivative which marks abrupt rises in energy with narrowed peaks (Bello et al., 2005; Eq. 2):

$$E_{(\mathrm{n})} = \frac{1}{N} \sum_{m=-N/2}^{(\mathrm{N}/2)-1} \left[x\left(n+m\right) \right]^2 w\left(m\right) \tag{2}$$

A direct method of detecting onsets arises from the derivative of the signal (1st or higher), which illuminate periods of change in the movement. The *onset of attack* would be the beginning of the periods where the 1st derivative is positive. The *onset of decay* corresponds with the end of the *attack,* in this

example, marked by the beginning of the period where the 1st derivative is negative. However, this naïve approach is still susceptible to the presence of noise (non-ideal sensors), overlapping participant responses (i.e., no return to baseline) or competing sources of spurious onsets (e.g., movement artefacts, movement harmonics).

4.1.3 Event Detection

Peak Picking involves a decision about candidate onsets (which are normally local maxima), resulting from the previous stage(s). If the ODF has been sufficiently well constructed, or the pre-processed data itself is suitable due to experimental design, this final stage is often simple thresholding. That is, candidate onsets that have a peak value above a certain threshold are considered to be movement onsets. This can be readily hand tuned in well-formed movement experiments. Other domain specific knowledge can be added, such as the expected recurrence of onsets within specific durations, a minimal/maximal duration etc.

We provide in the accompanying code, *peakdet*, one of the more robust peak detection algorithms written by Eli Billauer.[1] However, even the best algorithms are likely to have false or missed detections, again due to noise on the signal from imperfections in the sensor or due to human artefacts such as false movements. Therefore, onset detection methods are typically complemented with manual visual checks to ensure any errors are removed. We have further written a Matlab based graphical user interface to visually check the peak onsets extracted using the *peakdet* code, which accompanies this chapter.

To measure the effectiveness with each change made in the process, we need measures of performance. If we consider that merely capturing all of the movement onsets is not sufficient, we must also reject non-movement onsets, which gives rise to two measures: Sensitivity and Specificity. We define Sensitivity, also known as the true positive rate, as:

$$\text{Sensitivity (Recall)} = \text{Correct Movement Onset Detections} \,/\, \text{Total True Movement Onset Detections}$$

In which Total True Movement Onset Detections are the total number of true movement onsets detected + the number of missed onsets. We define the Positive Predictive Value (PPV), as:

$$\text{PPV} = \text{Correct Movement Onset Detections} \,/\, \text{Total Movement Onset Detections}$$

1 http://www.billauer.co.il/peakdet.html

In which Total Movement Onset Detections are the total number of true movement onsets + false onsets detected.

Trivially, we could have a Sensitivity 1, by setting the threshold below the lowest peak. All true movement onsets would be captured by this threshold. This would unfortunately minimise PPV, i.e., permit a maximal number of non-movement onsets to pass the threshold and be labelled falsely as movement onsets. There is thus a trade-off between the two values.

4.1.4 Dimensionality Reduction, Clustering and Machine Learning

The algorithms presented perform similarly to expert annotators' subjective agreement of the incidence of onsets in single channel data. Multiple marker systems can result in onset complexes, in one or more channels, coincident with a true movement onset. Onset complexes in isolation would correspond with an onset in a single channel system. These onset complexes require a further stage to evaluate when they become multi-channel features. Simple stages include considering one channel of data as representative of the whole (identical to prior mono channel approaches), or a sum of coincident onset complexes across channels compared to a threshold. Such a threshold may not capture the expert appraisal of multichannel cues that give rise to effective subjective onset labelling (e.g., in electromyography (EMG) contiguous channel onset complexes may result from electrodes associated with one muscle vs. multi-channel artefacts such as heartbeat contamination). Consistent labelling of multichannel onset complexes can be facilitated by: dimensionality reduction strategies (such as principal component analysis) and/or machine learning (clustering with an additional classification stage).

Principal component analysis (PCA) is a linear method of data re-expression which returns a set of n components, where n is equal (or less) to the dimension of the original data. These components are ordered by their explicative power of the variance, of the original signal. If the underlying movement is the greatest source of variance, then the principal component will be a single channel representative of the underlying movement. By focusing on that principal component, mono-channel strategies can be re-employed. Other methods of dimensionality reduction include independent component analysis (ICA) and multidimensional scaling. When lower dimensional expressions do not collapse to one obvious channel, i.e., suggesting multiple and/or non-linear underlying generators, machine learning methods can be applied (e.g., self-organising maps, generative topographic mapping).

Whilst Matlab has a Neural Network Toolbox at an extra cost, these approaches can be implemented in Matlab using the excellent NETLAB toolbox

(Nabney, 2002; http://www.aston.ac.uk/eas/research/groups/ncrg/resources/netlab/). NETLAB is also largely compatible with Octave.

4.2 *Extracting Onsets from Audio Recordings*

An audio signal contains distinct events pertaining to one or multiple acoustic sources. Examples include a sequence of musical notes, a chain of percussive hits, and consonant and vowel segments comprising continuous speech. Event timing information is conveyed through variation in some physical property of the source. These changes are detected by the listener and registered as distinct events that are often inter-connected at a higher contextual level. The task of extracting timing information about events embedded within an audio signal involves estimating perceptually important points of change. In particular, we are interested in detecting the presence of new acoustic events and annotating associated temporal information, (e.g., start time, end time and event duration). Most research in the field of audio signal processing targets the automatic detection of event onsets. Onset detection is highly relevant when studying the synchronisation in music performance where accurate measurement of response time is imperative.

As with movement (Section 4.1), the term *onset* is generally used to denote the earliest time at which a signal evolves quickly (Bello et al., 2005). This definition relates to the physical properties of the source and thus does not necessarily correlate with the perceived start of an event (Von & Rasch, 1981) or the perceived *attack* time which refers to the moment of rhythmic emphasis for a musical tone (Gordon, 1987; see Collins, 2006, for a review of modelling perceptual attack time and associated problems therein). Nevertheless, most recent work on music onset detection takes a pragmatic approach by tuning and assessing detection algorithms using hand-labelled datasets. Such ground-truth data is typically generated by experienced individuals who combine critical listening with spectro-temporal analysis using state-of-the-art software to best identify the beginning of acoustic events that satisfy the requirements of many practical applications.

Most onset detection algorithms deal with a monophonic signal corresponding to a *single* acoustic stream. The onset detection process follows the same principle as that described for movement onset detection (Section 4.1): Pre-processing, ODF transformation and finally event/feature extraction.

4.2.1 Signal Feature Based Detection Functions

The success of the system is fundamentally dependent on the reduction stage and so most effort has been on developing and evaluating different detection functions (Bello et al., 2005; Böck et al., 2012b; Collins, 2005a, Dixon, 2006). Perhaps the simplest of approaches to onset detection are those based on the

amplitude envelope (Masri, 1997; Schloss, 1985). The general idea is that the onset of a new sound leads to a sharp rise in the envelope of the waveform. The local energy of the signal can also be followed rather than the amplitude (Bello et al., 2004), for example by applying a running sum low-pass filter to the square of the signal. It is common to use the time derivative of the envelope such that significant changes in amplitude (or energy) are transformed to sharp peaks that are easily detected by thresholding the resulting detection function.

ODFs based on temporal features are generally adequate for percussive sounds and provide good temporal resolution and have low computational demand. Klapuri (1999) suggested taking the logarithm of the envelope prior to differencing to minimise spurious local maxima after the physical onset of the sound and emphasise lower intensity onsets. A further refinement is to incorporate spectral information since transients tend to introduce energy at high frequencies. The short-time Fourier transform (STFT) is commonly used for this purpose, although auditory filter banks have also been employed (Klapuri, 1999). Masri (1997) used the STFT to focus the local energy measurement towards high frequencies, a technique useful for emphasising the percussiveness of a sound. This high frequency content (HFC) detector can, however, be problematic for low-pitched and non-percussive instruments (Bello et al., 2005). In order to incorporate changes in the distribution of spectral energy over time, Masri (1997) proposed the spectral flux detector. Rather than summing the weighted magnitudes prior to differencing, the algorithm first sums over all positive changes in magnitude in each frequency bin between consecutive analysis frames generated by the STFT. Because changes in magnitude are measured across different frequency bands, the detection function is more reliable compared to one based solely on the temporal envelope.

Additional spectral methods make use of the phase spectra to enhance subtle tonal variations in the signal, and are less dependent on changes in energy (Bello et al., 2004). The idea is that during the steady-state portion of the signal, differences between the (unwrapped) phase of consecutive spectral frames will be constant. The phase deviation, defined as the second difference of the phase, i.e. the change in instantaneous frequency, can then be used to signify changes in the stationarity of the signal; large deviations are more probable during the attack region of a transient. Although methods incorporating phase information are better suited for sounds with soft onsets, one of the shortcomings of the phase deviation detector is its susceptibility to phase distortion and noise in low-energy components. Refined techniques include the weighted phase deviation and variations of the complex domain method, the latter combining both phase and magnitude information (Dixon, 2006; Duxbury et al., 2003). Finally, Collins (2005b) used the constant-Q pitch estimator (Brown & Puckette, 1993) as the primary feature driving an

onset detection algorithm targeting pitched non-percussive instruments. The algorithm incorporates vibrato suppression to better emphasise note transitions, outperforming the phase deviation algorithm of Bello et al. (2004)

Böck and Widmer (2013a) also proposed an onset detector with vibrato suppression, based on the common spectral flux method. The detector, called SuperFlux, uses a maximum filter applied to a logarithmic-frequency scaled spectrogram to better track spectral trajectories. The performance of SuperFlux outperformed the pitch-based detector of Collins (2005b) and another specialised detector targeting pitched non-percussive sounds (Schleusing et al., 2008). A second algorithm, the ComplexFlux, also based on differences in magnitude spectra was later developed (Böck & Widmer, 2013b) to suppress both vibrato and tremolo in solo pitched instruments. Figure 9.5 shows the temporal waveform and spectrogram of a recording of a violinist, playing with a détaché bowing style, from which four ODFs have been extracted. The signal was pre-processed by applying a 3rd order Butterworth high-pass filter to remove low-frequency noise picked up by the clip-on microphone. The simplest of detectors, which we have found to work well on signals with well-defined disconnected notes, is the Log HFC, obtained by applying the first-order difference to the logarithm of the frequency-weighted energy.

4.2.2 Classification Based Onset Detection

In recent years, machine learning techniques have been employed to overcome the issue of source-dependent onset detectors (Zhu et al., 2014) as well as establishing more sophisticated detection functions by learning directly from the human annotated datasets traditionally used to evaluate the aforementioned heuristic approaches (Davy & Godsill, 2002; Eyben et al., 2010; Lacoste & Eck, 2007; Marchi et al., 2014; Marolt et al., 2002; Toh et al., 2008). In general, the task is treated as a classification problem where spectral frames extracted from the audio signal are classified as being onsets or non-onsets. Supervised machine learning techniques such as Support Vector Machines (SVM) and Gaussian Mixture Models (GMM) have been employed (Kapanci & Pfeffer, 2004; Toh et al. 2008) to handle pitched non-percussive instruments such as the singing voice where "soft" onsets often occur between smooth pitch transitions and tend to be accompanied by complex modulations in pitch and amplitude.

Neural networks have proven successful in automatically locating onsets in a range of musical signals and define the current state-of-art (Böck et al., 2012a; Eyben et al., 2010; Marchi et al., 2014; Schluter & Böck, 2014). These methods use features such as cent-scaled magnitude spectrograms and linear prediction errors derived from multi-resolution spectra as inputs to a neural network which has been trained using binary labelled features to discriminate between

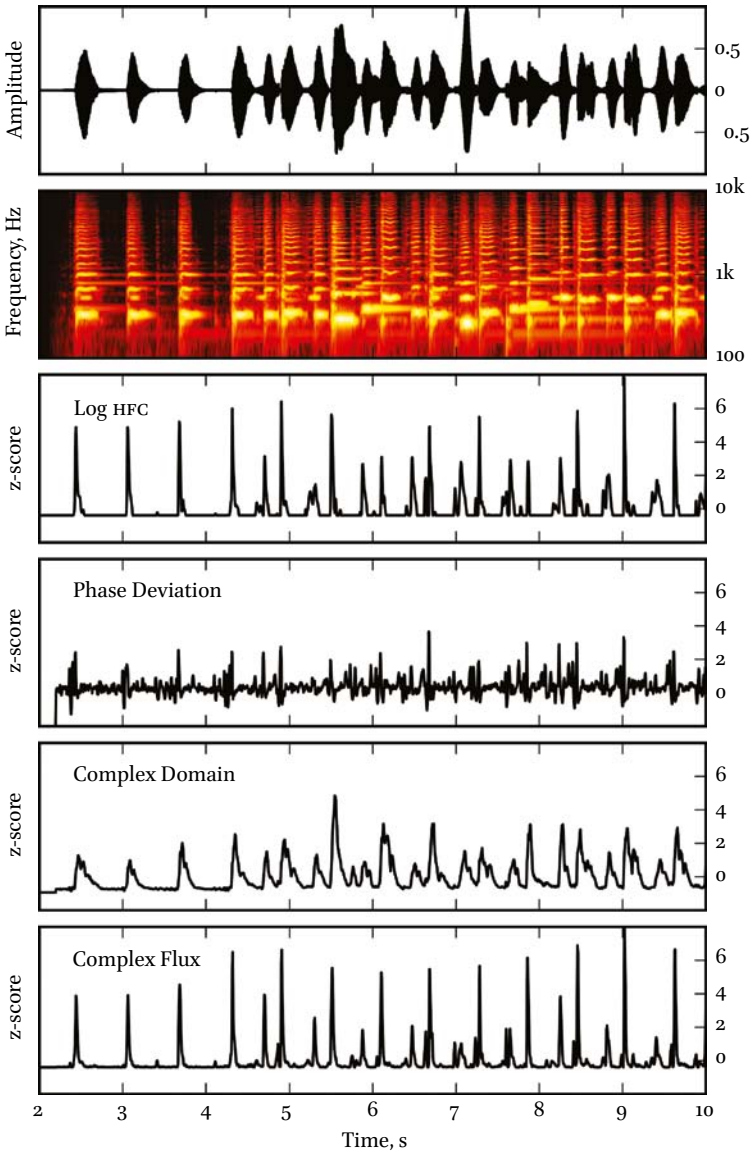

FIGURE 9.5 *Temporal waveform, log magnitude spectrogram and four onset detection functions (ODF) extracted from a violin recording. The ODFs have been standardised by setting their means to zero and standard deviations to one.*

onsets and non-onsets. Eyben et al. (2010) and Marchi et al. (2014) used long short-term memory (LSTM) models, a form of recurrent neural network (RNN) that provides complete access to past and future information over long time periods. These systems are therefore able to model the context in which onsets occur. Both systems were evaluated against traditional detection methods, (e.g., those presented by Bello et al. (2005) and Dixon (2006)), and showed superior performance with respect to F-measure (see Section 4.2.3), regardless of onset type and are therefore considered to be robust. Böck et al. (2012a) developed a real-time online version of the offline RNN method (Eyben et al., 2010), which although not as accurate, outperformed existing non-ML onset detectors (see also Böck et al., 2012b). Schluter and Böck (2013, 2014) enhanced their offline algorithm by replacing the RNN with a convolutional neural network (CNN), which requires less manual pre-processing and yields superior performance.

4.2.3 Performance and Considerations

Today's music onset detection methods are typically evaluated using human-annotated datasets of real-world acoustic sounds subdivided into classes based on instrument type. Although the manual annotation process is thorough and involves multiple assessment procedures performed by 3–5 experienced individuals, it is nonetheless subjective, thus blurring the distinction between physical onset and perceptual onset. Because of this uncertainty, detected onsets are deemed valid if within 50 ms of the subjective position (Bello et al., 2005), although a lower tolerance of 25 ms has been used by some authors (Böck et al., 2012b), especially for percussive sounds where physical onsets are well-defined (Collins, 2005a). Similar to those described in section 4.1.3, standard evaluation metrics include precision (P), recall (R) and F-measure (F), defined respectively in Eq. 3–5.

$$P = \frac{O_c}{O_c + O_{fp}} \tag{3}$$

$$R = \frac{O_c}{O_c + O_{fn}} \tag{4}$$

$$F = 2\frac{PR}{P + R} \tag{5}$$

Here, O_c is the number correctly detected onsets, O_{fp} is the number of false positives and O_{fn} is the number of false negatives. In offline settings one might favour high recall over precision, since there is greater chance that the detector

is simply reacting to noise or modulations in the signal unrelated to onsets. In this respect, it is less subjective to manually remove data points than to add them.

Table 9.1 gives the average F-measures by instrument category for four of the best onset detectors submitted the 2015 onset detection contest run by the Music Information Retrieval Evaluation eXchange[2] (MIREX). Both Universal (Böck et al., 2015; Eyben et al., 2010) and Fusion (Chen, 2015) algorithms use probabilistic methods, whereas SuperFlux (Böck & Widmer, 2013a) and ComplexFlux (Böck & Widmer, 2013b) are refined versions of two classic spectral-based algorithms. As with the MIREX results, The detectors are ranked by the average of their class means, though we have omitted the third best ranking algorithm as it is an online version of the Universal detector (Böck et al., 2012a). For this dataset, the four techniques perform similarly over all classes (around 80%) but there are clear differences between the algorithms within each category. For example, the probabilistic methods outperform the simpler flux algorithms in the majority of classes with a few exceptions (e.g., solo singing voice). This may be attributable to a lack of training data and/or because the flux detectors – especially ComplexFlux – were designed to better handle instruments with strong vibrato and tremolo. For all algorithms, performance appears to deteriorate for the voice, sustained strings and wind instruments,

TABLE 9.1 *Summary of average F-measures for four state-of-the-art onset detectors submitted to 2015 MIREX audio onset detection contest.*

Class	**Universal**	**Fusion**	**Complex-Flux**	**Super-Flux**	**Mean (SD)**
Complex	79.4	79.5	75.7	77.5	78.0 (1.8)
Poly pitched	94.1	93.9	91.7	91.6	92.8 (1.4)
Solo bars & bells	100.0	100.0	96.5	96.7	98.3 (2.0)
Solo brass	82.1	77.0	75.3	76.6	77.8 (3.0)
Solo drum	93.1	93.1	93.1	92.4	92.9 (0.3)
Solo plucked strings	90.9	91.5	89.7	89.8	90.5 (0.8)
Solo singing voice	52.1	55.3	60.4	60.6	57.1 (4.1)
Solo sustained strings	72.9	66.9	57.5	58.8	64.0 (7.2)
Solo winds	74.0	72.2	74.6	68.6	72.3 (2.7)
Mean	82.1	81.0	79.4	79.2	
(SD)	(13.8)	(13.9)	(13.4)	(13.5)	

2 http://www.music-ir.org/mirex/wiki/2015:Main_Page

which might be explained by difficulties in detecting softer onsets and/or that the human annotations were more in line with perceptual attack time rather than the physical onsets picked up by the algorithms. The average recall (and precision) across all test stimuli for the four algorithms was: Universal – 87.9% (86.2%); Fusion – 86.2% (87.0%); ComplexFlux – 84.3% (86.8%); SpectralFlux – 85.6% (85.4%).

The choice of algorithm for detecting sonic events is evidently dependent on the both source type of the technical requirements of a given application. When studying synchronisation in musical performance, the measurement of player timing must be sufficiently accurate to reflect the tempo of the piece and capture salient asynchronies between the note onsets of each player and those of an external auditory stimulus such as a metronome and/or the note played by respective partners. For example, capturing small asynchronies in timing is imperative when studying how performers correct for deviations from an external beat (Vorberg & Wing, 1996) or from fellow musicians (Rasch, 1979, Wing et al., 2014), or how players utilise asynchrony for expressive purposes (Palmer, 1996). In general, methods based on amplitude envelope following provide the highest temporal resolution and are computationally efficient compared to frequency-domain and especially ML approaches. The latter are more suitable for acoustic sources with soft attacks and complex modulations following the onset, such as those produced by bowed string instruments, flute or the singing voice. When using frequency-domain methods, it is important to consider the parameters used to configure the time-frequency decomposition, such as window length and window hop size in the case of the STFT. For example, reducing the window hop size improves temporal precision at the cost of increasing the workload and smoothing variations in the resulting detection function. The choice of window size, which defines the temporal resolution, is signal-dependent and therefore multi-resolution analysis is more favourable in the case of complex signals.

In short, it is preferable to employ an offline onset detector, which, along with the peak-picker, can be tuned for maximum accuracy. With sufficient training data, probabilistic multi-resolution methods are robust, but one should be cautious of the quality of the subjective data used to train the classification. For more objective measures of onset, the flux methods can be chosen and combined with other detectors to increase the likelihood of capturing new events based on changes across multiple signal features. Most state-of-the-art onset detectors output onset times to text files which, along with the audio signal, can be imported to audio analysis software such as Sonic Visualiser (Cannam et al., 2010) for cross-validation using displays of spectrograms and other signal features, and vari-speed playback.

5 Analysis and Modelling

5.1 *Alignment*

Once the challenge of accurate onset or event detection has been achieved, the standard measures of analysis are relatively simple. However, to analyse synchrony between the events of two or more independent sources (e.g., the heel strikes of a group of people walking or the auditory onsets of a string ensemble) the onsets need to be aligned to understand which event from one source is temporally related to an event from the other source(s). Here we describe an approach to pairwise-align response onsets to the cue onsets (Elliott et al., 2009b). Note, that typically for multi-person alignment, there is ideally a common vector of 'cue' onsets to which all the response onsets from each group member can be aligned to. This may be an external metronome stimulus or the movements of the lead person in the group, for example.

In this approach we use a dynamic programming method to find the shortest distance between response and cue onsets (Figure 9.6). Starting with two vectors, one containing the 'cue' onsets, $\boldsymbol{m}$ (this could be another person's movement onsets, or a fixed stimulus such as a metronome); the other containing the response onsets, $\boldsymbol{t}$. The length of $\boldsymbol{m}$ and $\boldsymbol{t}$ do not need to be equal. We subsequently make a matrix of squared distances, $\boldsymbol{d}$, between each cue onset and each response onset. Alignment occurs by matching up each response onset to the closest (i.e., shortest squared distance) cue onset. If a cue onset

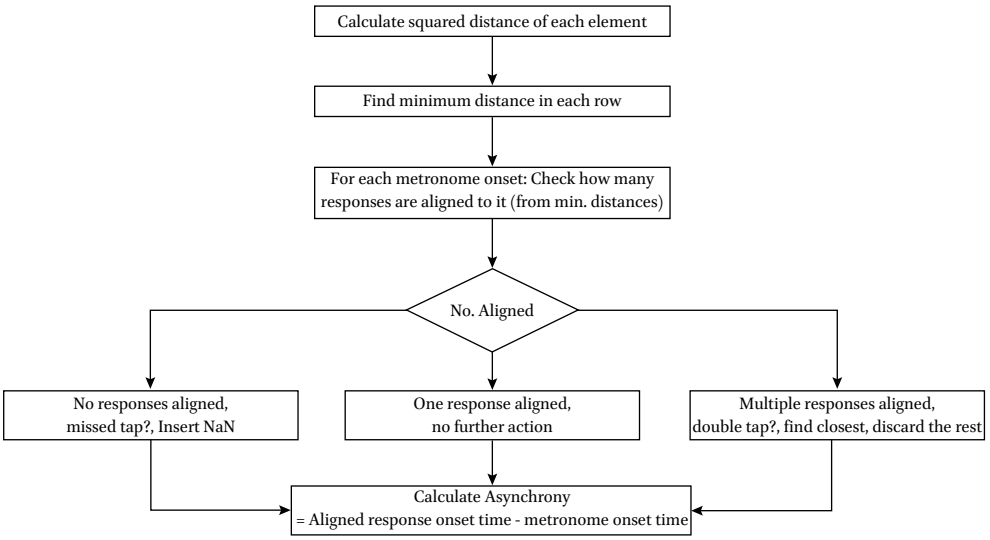

FIGURE 9.6 *Flowchart of the algorithm used to align onsets between two sources (taken from Elliott et al., 2009b).*

has no matching response, then a *NaN* is inserted. If a cue onset has multiple matching responses, we first check to see if the previous onset is empty (*NaN*). If so, we assign the earlier response to the previous cue onset. Of the remainder we assign the response, which has the shortest distance to that cue onset.

5.2 *Calculating Asynchrony and IOI/ISI*

Once the onsets of the participant(s) have been aligned to either an external cue or other participant onsets, calculating the time difference or asynchrony (*A*; Eq. 6) between related onsets and the IOIs (also labelled as inter-stimulus intervals, *ISI* for cue onset intervals; Eq. 7) is relatively trivial.

$$A_k = t_k - m_k \tag{6}$$

$$IOI_k = t_{k+1} - t_k \tag{7}$$

where t_k is the k th response onset and m_k is the k th cue onset.

5.3 *Participant Outliers*

Sensorimotor synchronisation analysis can be very sensitive to outliers. Outliers will generally emerge in the latter stages of analysis, where the IOI or the asynchronies have been calculated. For example, a missed movement onset to a cue stimulus with an interval of 500ms, will suddenly introduce an IOI of 1000 ms. Another common issue, occurs when someone's movement onset occurs very late, or very early, relative to the comparative cue onset. This will result in a phase wrapped asynchrony (e.g., one that is assigned, via alignment, as a late response to the preceding cue onset rather than an early response to the current cue, or vice-versa). Both these occurrences will result in large within-trial standard deviations (SD) emerging for the IOI and asynchronies, respectively. In fact, it is useful to become familiar with the range of IOI and/or asynchrony SD you would expect from a 'good' trial. This helps to spot potentially erroneous trials during analysis. As an example, for a simple finger tapping task to an auditory metronome with an ISI of 500 ms, one would expect the both the IOI and asynchrony SD to be in the range of 15–30 ms. Values far exceeding this range suggest the trial should be examined in more detail.

Outlier removal must be dealt with methodologically and consistently. Using the IOIs to find outliers is often the simplest and most robust method. Working with asynchronies is much more challenging. A robust approach for detecting IOI outliers is the inter-quartile range method. In Matlab, using the *median* command to find the median IOI of the trial data and then the *iqr*

command to find the inter-quartile range, the threshold for upper and lower outlier values is:

$$\text{threshUpper} = \text{median}(\text{IOI}) + \text{N}^*\text{iqr}(\text{IOI});$$
$$\text{threshLower} = \text{median}(\text{IOI}) - \text{N}^*\text{iqr}(\text{IOI});$$

Where *IOI* is a vector of IOIs calculated from a trial. *N* defines how 'strong' the outlier detection function is. *N*=3 should be the minimum and will heavily cleanse the data, while *N*=6 will be more conservative and only remove extreme outliers.

Matlab's *find* command can subsequently be used to locate values exceeding either the upper or lower threshold and can be removed from the IOI vector. Removal will usually consist of replacing the value with *NaN* and henceforth using *nanmean*, *nanstd* is required to calculate the mean and SD of the cleansed IOIs. However, if cross-correlation or other calculations relying on a continuous series are to be applied, then an alternative replacement method should be used (e.g. replacing with the average (median) value or similar).

The removal/replacement of IOIs should be reflected in the corresponding Asynchrony vector. Assuming *IOI* and *A* are calculated as defined in Eq. 6–7, then removal of IOI_k should result in removal of A_{k+1}.

Identifying outliers from asynchrony data is more challenging. Recall, the alignment process will allocate a response onset to a cue onset within the range –ISI/2 to ISI/2, where ISI is the inter-stimulus interval of the cue. Therefore, given all asynchronies will be bound within this range, there are no outliers as such. However, if phase wrapping occurs there will be sign changes where the onsets go from being large and negative to large and positive, or vice versa. This corresponds to drift (see Figure 9.7) where the participant is not synchronising with the cue and therefore asynchronies become increasingly negative until they hit the lower bound and subsequently the next response is closer to previous cue onset but with a positive asynchrony.

There is little that can be done with linear analyses in these scenarios. The SD becomes very large when these discontinuities occur. And given that typically the presence of drift suggests the participant isn't synchronising to the cue, it is often a case of discarding trials where this occurs. There are occasions where phase-wrapping is likely and of interest (e.g., analysing data where the response and cue have differing tempos). In these cases, it is recommended that circular statistics be used to analyse the data. Circular mean and SD can be used as an alternative without being susceptible to the phase wrapping discontinuities. Further details of circular statistics

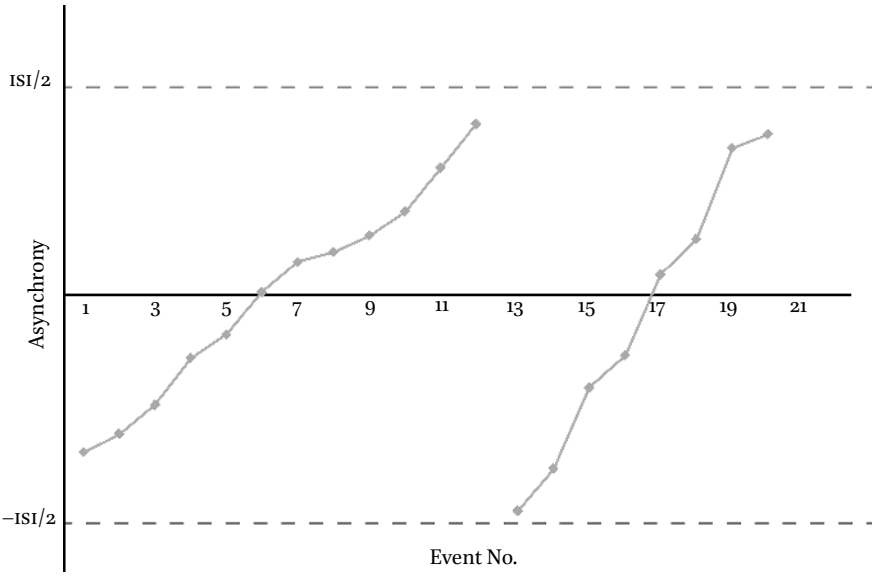

FIGURE 9.7 *Typical pattern of asynchronies when participant is exhibiting drift – i.e. not synchronising with the cue. Note the phase wrapping occurs around +/– ISI/2 and can result in highly inflated asynchrony variance.*

are beyond the scope of this chapter but an excellent free toolbox for Matlab is available (Berens, 2009; http://www.mathworks.com/matlabcentral/fileexchange/10676-circular-statistics-toolbox--directional-statistics-).

5.4 *Cross- and Auto-Covariance and Event Based Synchronisation Models*

It is often useful to measure the auto-covariance of the IOIs and asynchronies, or the cross-covariance between two IOI / asynchrony time series. The auto-covariance shows the dependency between current and past time-series values. Wing and Kristofferson (1973) proposed a model for tapping without an external stimulus that predicts that finger tapping intervals have a lag -1 dependence (resulting in a short-long-short-long pattern). The model follows the hypothesis that tapping is based on two internal processes: time keeping (that maintain a temporal interval) and a motor action (that is a result of the execution of a given motor command). The model can be written as:

$$IOI_k = T_k + M_{k+1} - M_k \tag{8}$$

Where, IOI_k is the kth IOI (see Eq. 7) and T_k and M_k are the timekeeper interval and motor delay respectively.

The model predicts that:

$$\gamma_{\mathrm{IOI}}(1) = -\sigma_T{}^2 \tag{9}$$

$$\gamma_{\mathrm{IOI}}(0) = \sigma_T{}^2 + 2\sigma_M{}^2 \tag{10}$$

Where $\gamma_I(k)$ is the lag k auto-covariance, $\sigma_T{}^2$ is the timekeeper variance, $\sigma_M{}^2$ is motor variance.

The model has become a highly efficient tool to characterise tapping (Wing, 2002). Empirical results indicate several intriguing relations between timekeeper variance, motor variance, and tempo. Namely: (a) Motor noise remains constant when the base tapping tempo is changed, but timekeeper variance increases with tempo; (b) Motor noise is smaller than timekeeper noise $\left(\sigma_M{}^2 < \sigma_T{}^2\right)$

The success of the model led to its generalisation to the case of tapping to an external metronome. Vorberg and Wing (1996) proposed a revised model which included a correction gain parameter to describe the process of synchronising to an external cue. The correction gain, α (often also referred to as phase correction), explains how much of the previous error (asynchrony) is corrected for in the next movement.

$$IOI_k = \alpha A_k + T_k + M_{k+1} - M_k \tag{11}$$

The gain is stable in the range, $0 \le \alpha \le 2$, where $\alpha = 1$ is full correction, $\alpha > 1$ is overcorrection, and $\alpha < 1$ is undercorrection. In most cases, empirical estimates of α are usually in the range of 0.5 to 1.

In the case of a relatively stable metronome (no significant tempo changes), the correction gain can be deduced simply by calculating the cross-covariance between the cue and response intervals, *if* the cue intervals do not have zero variance (i.e., an isochronous metronome has zero interval variance). The relationship between the covariance and the correction gain is as follows:

$$\gamma\_CI(j) = \alpha(1-\alpha)\wedge(j-1)\sigma\_C\wedge 2, j \ge 0 \tag{12}$$

where, $\gamma_{CI(j)}$ is the cross-covariance function between the stimulus response intervals of lag j and $\sigma_c{}^2$ is the variance of the stimulus intervals. The model of Eq. 11 can be generalised to ensemble synchronisation:

$$IOI_k = \sum_i \alpha_i A_{k,i} + T_k + M_{k+1} - M_k \tag{13}$$

Where $A_{k,i}$ are the asynchronies between the studied player all other players, and α_i is the phase correction parameter associated with adapting to a specific player i.

Note that this model is a generalisation of an ensemble synchronisation model proposed by (Wing, Endo, Bradbury, & Vorberg, 2014) that has been used to study synchronisation within string quartets. Their model is identical to Eq. 13, but with the assumption that the parameter $\sigma_M{}^2 = 0$, and therefore $M_{k+1} - M_k$ in equation Eq. 13 does not play a role.

It is possible to generalise the model of equation Eq. 11 to the case where there are substantial tempo changes (Schulze, Cordes, & Vorberg, 2005). Here it is often assumed that an additional period correction process occurs (Repp & Keller, 2008):

$$\tau_k = \tau_{k-1} - \beta A_k \tag{14}$$

Where $\tau_k = mean(T_k)$ and β represents the period correction constant.

The model can be cast as a standard Autoregressive Moving Average (ARMA) model (Diedrichsen, Ivry, & Pressing, 2003)

$$IOI_k - IOI_{k-1} = -(\alpha + \beta)A_k + \alpha A_{k-1} + T_k - T_{k-1} + M_{k+1} - 2M_k + M_{k-1} \tag{15}$$

Here T_k and M_k are two independent random variables with a fixed mean.

5.5 *Bounded General Least Squares (bGLS) Method for Parameter Estimation*

While the cross- and auto-correlation approaches to parameter estimation of the linear phase correction model are relatively simple to compute, their application is limited to one participant with small tempo variations. In the case of ensemble synchronisation, the estimation procedure based on the autocovariance function requires a slow iterative model fitting approach.

Moreover, recent work (Jacoby, Keller, Repp, Ahissar, & Tishby, 2015) showed that the structure of the models described above generates an inherited dependency in the accuracy of estimating the parameters α, $\sigma_M{}^2$ and $\sigma_T{}^2$. Since the parameters are inherently interdependent, they cannot be jointly estimated by the autocovariance method or by *any other method* without using further assumptions. Therefore, directly applying the autocovariance method or standard linear estimation techniques such as the Matlab *armax* command on data will often lead to unreliable estimations. While the problem exists also for single participant synchronisation with a metronome with small tempo changes, it becomes much more notable in the case of ensemble synchronisation or when there are large tempo changes. Fortunately, there is

a simple solution to this problem. As mentioned above it has been empirically observed that the motor noise is smaller than timekeeper variance. If this additional simple constraint is taken into account in the estimation process the interdependency problem is practically resolved (Jacoby et al., 2015). Moreover, the group further proposed an algorithm, the bounded general least squares (bGLS) method that can estimate the parameters in case of a single person and ensemble synchronisation as well as tempo changing sequences (Jacoby, Tishby, Repp, Ahissar, & Keller, 2015).

The method has been applied in music related studies, re-analysing the earlier quartet study by Wing et al., (Jacoby et al., 2015) and investigating metrical structure in Malian jembe drumming (Polak, Jacoby, & London, 2015). In addition, it has been applied to a group sensorimotor synchronisation task in order to estimate changes in correction along a chain of individuals moving in time with each other (Honisch et al., 2016).

The mathematical derivations of the method are fully explained in the two aforementioned publications (Jacoby et al., 2015) and, hence, won't be reiterated here (for a short overview of the method see Elliott, Chua & Wing, 2016). However, the bGLS Toolbox for Matlab is provided with example code for this chapter (see book's GitHub repository).

6 Conclusion

We have presented methods for collecting, conditioning and analysing the timing of movements, ranging from simple finger tapping where response events can be captured by switches, force transducers or motion capture systems to the complexities of music performance where the data commonly requires acoustic recording, or in some cases, motion capture data. Regardless of the particular technology for capturing timing data, our goal has been to maximise the measurement accuracy in order to better characterise, not only the accuracy of timing in terms of mean and variability, but also the form of variability, in order to reveal the underlying mechanisms that are so often key to the skilled performance of complex sequential activities such as music and dance.

References

Balasubramaniam, R., A.M. Wing, & A. Daffertshofer (2004). Keeping with the beat: Movement trajectories contribute to movement timing. *Experimental Brain Research*, *159*(1), 129–134.

Bello, J.P., L. Daudet, S. Abdallah, C. Duxbury, M. Davies, M.B. Sandler, & S. Member (2005). A tutorial on onset detection in music signals. *IEEE Transactions on Speech and Audio Processing*, *13*(5), 1–13.

Bello, J.P., C. Duxbury, M. Davies, M. Sandler, & S. Member (2004). On the use of phase and energy for musical onset detection in the complex domain. *IEEE Signal Processing Letters*, *11*(6), 553–556.

Berens, P. (2009). CircStat: A MATLAB toolbox for circular statistics. *Journal of Statistical Software*, *31*(10).

Böck, S., & G. Widmer (2013a). Maximum filter vibrato suppression for onset detection. In *16th International Conference on Digital Audio Effects* (*DAFx-13*).

Böck, S., & G. Widmer (2013b). Local group delay based vibrato and tremolo suppression for onset detection. In *Proceedings of the 13th International Society for Music Information Retrieval Conference* (*ISMIR*).

Böck, S., A. Arzt, F. Krebs, & M. Schedl (2012a). Online real-time onset detection with recurrent neural networks. In *15th International Conference on Digital Audio Effects* (DAF*x*).

Böck, S., F. Krebs, & M. Schedl (2012b). Evaluating the online capabilities of onset detection methods. In *13th International Society for Music Information Retrieval Conference* (*ISMIR*) (pp. 49–54).

Böck, S., F. Krebs, F. Korzeniowski, & G. Widmer (2015). *MIREX 2015 Submissions*. MIREX Audio Onset Detection. http://nema.lis.illinois.edu/nema_out/mirex2015/results/aod/.

Brown, J.C., & M.S. Puckette (1993). A high resolution fundamental frequency determination based on phase changes of the fourier transform. *Journal of the Acoustical Society of America*, *94*(2).

Cannam, C., C. Landone, & M. Sandler (2010). Sonic visualiser: An open source application for viewing, analysing, and annotating music Audio Files. In *Proceedings of the ACM Multimedia 2010 International Conference* (pp. 1467–1468).

Chen, C. (2015.). *An improved onset detection algorithm by ODF fusion*. MIREX Audio Onset Detection. http://nema.lis.illinois.edu/nema_out/mirex2015/results/aod/.

Collins, N. (2005a). A comparison of sound onset detection algorithms with emphasis on psychoacoustically motivated detection functions. In *Proceedings of the 118th Audio Engineering Society Convention*.

Collins, N. (2005b). Using a pitch detector for onset detection. In *Proceedings of the International Symposium on Music Information Retrieval* (pp. 100–106).

Collins, N. (2006). Investigating computational models of perceptual attack time. In *Proceedings of the 9th International Conference on Music Perception & Cognition* (*ICMPC*).

Davy, M., & S. Godsill (2002). Detection of abrupt spectral changes using support vector machines an application to audio signal segmentation. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing* (ICASSP) (pp. 1313–1316).

De Poli, G., A. Roda, & A. Vidolin (1998). Note-by-note analysis of the influence of expressive intentions and musical structure in violin performance. *Journal of New Music Research*, *27*(3), 293–321.

Diedrichsen, J., R.B. Ivry, & J. Pressing (2003). Cerebellar and basal ganglia contributions to interval timing. In Meck, W.H. (Ed.), *Functional and neural mechanisms of interval timing*. CRC Press.

Dixon, S. (2006). Onset detection revisited. In *Proceedings of the International Conference on Digital Audio Effects* (DAF*x*).

Duxbury, C., J.P. Bello, M. Davies, & M. Sandler (2003). Complex domain onset detection for musical signals. In *Proceedings of the 6th International Conference on Digital Audio Effects* (DAF*x*).

Elliott, M.T., W.L. Chua, & A.M. Wing (2016). Modelling single-person and multi-person event-based synchronisation. *Current Opinion in Behavioral Sciences*, *8*, 167–174.

Elliott, M.T., A.E. Welchman, & A.M. Wing (2009a). Being discrete helps keep to the beat. *Experimental Brain Research*, *192*(4), 731–737.

Elliott, M.T., A.E. Welchman, & A.M. Wing (2009b). MatTAP: A MATLAB toolbox for the control and analysis of movement synchronisation experiments. *Journal of Neuroscience Methods*, *177*(1), 250–257.

Elliott, M.T., A.M. Wing, & A.E. Welchman (2010). Multisensory cues improve sensorimotor synchronisation. *European Journal of Neuroscience*, *31*(10), 1828–1835.

Elliott, M.T., A.M. Wing, & A.E. Welchman (2014). Moving in time: Bayesian causal inference explains movement coordination to auditory beats. *Proceedings of the Royal Society B: Biological Sciences*, *281*(1786), 20140751.

Ellis, M.C. (1991). An analysis of swing subdivision and asynchronization in three jazz saxophonists. *Perceptual and Motor Skills*, *73*(3), 707–713.

Eyben, F., S. Böck, B. Schuller, & A. Graves (2010). Onset detection with bidirectional long short-term memory neural networks. In *Proceedings Annual Meeting of the MIREX 2010 community as part of the 11th International Conference on Music Information Retrieval* (pp. 589–594).

Finney, S.A. (2001). FTAP: A Linux-based program for tapping and music experiments. *Behavior Research Methods, Instruments, & Computers*, *33*, 65–72.

Friberg, A., & A. Sundströöm (2002). Swing ratios and ensemble timing in jazz performance: Evidence for a common rhythmic pattern. *Music Perception: An Interdisciplinary Journal*, *19*(3), 333–349.

Georgiou, L., V. Racic, J.M.W. Brownjohn, & M.T. Elliot (2015). Coordination of groups jumping to popular music beats. In Caicedo, J. & S. Pakzad (Eds.), *Dynamics of Civil Structures, Volume 2* (pp. 283–288). Springer International Publishing.

Goebl, W., & C. Palmer (2008). Tactile feedback and timing accuracy in piano performance. *Experimental Brain Research*, *186*(3), 471–479.

Goebl, W., & C. Palmer (2009). Synchronization of timing and motion among performing musicians. *Music Perception: An Interdisciplinary Journal*, *26*(5), 427–438.

Goebl, W., S. Flossmann, & G. Widmer (2010). Investigations of Between-hand Synchronization in Magaloff's Chopin. *Computer Music Journal*, *34*(3), 35–44.

Gordon, J.W. (1987). The perceptual attack time of musical tones. *The Journal of the Acoustical Society of America*, *82*(1), 88–105.

Hennig, H. (2014). Synchronization in human musical rhythms and mutually interacting complex systems. *Proceedings of the National Academy of Sciences*, *111*(36), 12974–12979.

Honisch, J.J., M.T. Elliott, N. Jacoby, & A.M. Wing (2016). Cue properties change timing strategies in group movement synchronisation. *Scientific Reports*, *6*.

Honisch, J.J., N. Roach, & A.M. Wing (2009). Movement synchronization to a virtual dancer: How do expert dancers adjust to perceived temporal and spatial changes whilst performing ballet versus abstract dance sequences. In *Proceedings of the ISSP 12th World Congress of Sport Psychology*.

Jacoby, N., P.E. Keller, B.H. Repp, M. Ahissar, & N. Tishby (2015a). Lower bound on the accuracy of parameter estimation methods for linear sensorimotor synchronization models. *Timing & Time Perception*, *3*(1–2), 32–51.

Jacoby, N., N. Tishby, B.H. Repp, M. Ahissar, & P.E. Keller (2015b). Parameter estimation of linear sensorimotor synchronization models: Phase correction, period correction, and ensemble synchronization. *Timing & Time Perception*, *3*(1–2), 52–87.

Kapanci, E., & A. Pfeffer (2004). A hierarchical approach to onset detection. In *Proceedings of the International Computer Music Conference*.

Keele, S.W., R.A. Pokorny, D.M. Corcos, & R. Ivry (1985). Do perception and motor production share common timing mechanisms: A correlational analysis. *Acta Psychologica*, *60*(2), 173–191.

Keller, P.E., G. Knoblich, & B.H. Repp (2007). Pianists duet better when they play with themselves: On the possible role of action simulation in synchronization. *Consciousness and Cognition*, *16*(1), 102–111.

Klapuri, A. (1999). Sound onset detection by applying psychoacoustic knowledge. In *Proceedings of The IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*.

Lacoste, A., & D. Eck (2007). A supervised classification algorithm for note onset detection. *EURASIP Journal on Advances in Signal Processing*.

Lago, N.P., & F. Kon (2004). The quest for low latency. In *Proceedings of the International Computer Music Conference* (*pp. 33–36*).

Loehr, J.D., & C. Palmer (2007). Cognitive and biomechanical influences in pianists' finger tapping. *Experimental Brain Research, 178*(4), 518–528.

Manning, F., & M. Schutz (2013). "Moving to the beat" improves timing perception. *Psychonomic Bulletin & Review*, *20*(6), 1133–1139.

Marchi, E., G. Ferroni, F. Eyben, L. Gabrielli, S. Squartini, & B. Schuller (2014). Multi-resolution linear prediction based features for audio onset detection with bidirectional LSTM neural networks. *2014 IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*).

Marolt, M., A. Kavcic, M. Privosnik, & S. Divjak (2002). On detecting note onsets in piano music. In *Proceedings of the 11th IEEE Mediterranean Electrotechnical Conference* (pp. 385–389).

Masri, P. (1997). Computer modelling of sound for transformation and synthesis of musical signals. Ph.D Thesis. University of Bristol.

Moore, G.P., & J. Chen (2010). Timings and interactions of skilled musicians. *Biological Cybernetics*, *103*(5), 401–414.

Nabney, I. (2002). *NETLAB: Algorithms for Pattern Recognition*. Springer Science & Business Media.

Palmer, C. (1996). On the assignment of structure in music performance. *Music Perception*, 14, 23–56.

Palmer, C., & J.C. Brown (1991). Investigations in the amplitude of sounded piano tones. *The Journal of the Acoustical Society of America*, *90*(1), 60–66.

Pecenka, N., & P.E. Keller (2011). The role of temporal prediction abilities in interpersonal sensorimotor synchronization. *Experimental Brain Research*, *211*(3–4), 505–515.

Pelton, T.A., A.M. Wing, D. Fraser, & P. Vliet (2015). Differential effects of parietal and cerebellar stroke in response to object location perturbation. *Frontiers in Human Neuroscience*, *9*.

Pfordresher, P., & C. Palmer (2002). Effects of delayed auditory feedback on timing of music performance. *Psychological research*, *66*(1), 71–79.

Polak, R., & J. London (2014). Timing and meter in mande drumming from Mali. *Music Theory Online*, *20*(1).

Polak, R., N. Jacoby, & J. London (2015). Ensemble entrainment in jembe music from Mali. *15th Rhythm Production and Perception Workshop, Netherlands, July 6–8, 2015*.

Rasch, R.A. (1979). Synchronization in performed ensemble music. *Acta Acustica united with Acustica*, *43*(2), 121–131.

Repp, B.H. (1995). Quantitative effects of global tempo on expressive timing in music performance: Some perceptual evidence. *Music Perception: An Interdisciplinary Journal*, *13*(1), 39–57.

Repp, B.H. (1999). Detecting deviations from metronomic timing in music: Effects of perceptual structure on the mental timekeeper. *Perception & Psychophysics*, *61*(3), 529–548.

Repp, B.H. (2005). Sensorimotor synchronization: A review of the tapping literature. *Psychonomic Bulletin & Review*, *12*(6), 969–992.

Repp, B.H., & P.E. Keller (2008). Sensorimotor synchronization with adaptively timed sequences. *Human Movement Science*, *27*(3), 423–456.

Repp, B.H., & Y.-H. Su (2013). Sensorimotor synchronization: A review of recent research (2006–2012). *Psychonomic Bulletin & Review*, *20*(3), 403–452.

Rosao, C., R. Ribeiro, & D.M. De Matos (2012). Influence of peak selection methods on onset detection. In *ISMIR* (pp. 517–522).

Schleusing, O., B. Zhang, & Y. Wang (2008). Onset detection in pitched non-percussive music using warping-compensated correlation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*).

Schluter, J., & S. Böck (2013). Musical onset detection with convolutional neural networks. In *Proceedings of the 6th International Workshop on Machine Learning and Music* (*MML*).

Schluter, J., & S. Böck (2014). Improved musical onset detection with convolutional neural networks. *IEEE International Conference on Acoustic, Speech and Signal Processing* (*ICASSP*).

Schultz, B.G., & F.T. Vugt. (2015). Tap Arduino: An Arduino microcontroller for low-latency auditory feedback in sensorimotor synchronization experiments. *Behavior Research Methods*, 1–17.

Schulze, H.H., A. Cordes, & D. Vorberg (2005). Keeping synchrony while tempo changes: Accelerando and ritardando. *Music Perception*, *22*(3), 461–477.

Shaffer, L.H. (1984). Timing in solo and duet piano performances. *The Quarterly Journal of Experimental Psychology*, *36*(4), 577–595.

Stables, R., S. Endo, & A. Wing (2014). Multi-Player microtiming humanisation using a multivariate markov model. In *International Conference on Digital Audio Effects* – DAF*x 14* (pp. 109–114).

Stevens, L.T. (1886). On the time sense. *Mind*, *11*, 393–404.

Thaut, M.H., B. Tian, & M.R. Azimi-Sadjadi (1998). Rhythmic finger tapping to cosine-wave modulated metronome sequences: Evidence of subliminal entrainment. *Human Movement Science, 17*(6), 839–863.

Toh, C.C., B. Zhang, & Y. Wang (2008). Multiple-feature fusion based onset detection for solo singing voice. In *Proceedings of the 9th International Conference on Music Information Retrieval* (*ISMIR*) (pp. 515–520).

Vincent, E. (2006). Musical source separation using time-frequency source priors. *IEEE Transactions on Audio, Speech, and Language Processing*, *14*(1), 91–98.

Vorberg, D., & A.M. Wing (1996). Modeling variability and dependence in timing. In *Handbook of perception and action* (pp. 181–262). London: Academic Press.

Vos, J., & R. Rasch (1981). The perceptual onset of musical tones. *Perception & Psychophysics*, *29*(4), 323–335.

Wang, Y., R. Stables, & J Reiss. (2010). Audio latency measurement for desktop operating systems with onboard soundcards. In *Audio Engineering Society Convention 128*. Audio Engineering Society.

Wing, A.M. (2002). Voluntary timing and brain function: An information processing approach. *Brain and Cognition*, *48*(1), 7–30.

Wing, A.M., & A.B. Kristofferson (1973). Response delays and the timing of discrete motor responses. *Perception and Psychophysics*, *14*, 5–12.

Wing, A.M., S. Endo, A. Bradbury, & D. Vorberg (2014). Optimal feedback correction in string quartet synchronization. *Journal of The Royal Society Interface*, *11*(93), 20131125.

Winter, D.A. (2009). *Biomechanics and Motor Control of Human Movement*. John Wiley & Sons.

Zhu, B., J. Gan, J. Cai, Y. Wang, & H. Wang (2014). Adaptive Onset Detection based on Instrument Recognition. In *Proceedings of the 12th International Conference on Signal Processing* (*ICSP*) (pp. 2416–2421).